DOCUMENT RESUME

ED 428 098 TM 029 499

AUTHOR van der Linden, Wim J.
TITLE Optimal Assembly of Tests with Item Sets. Research Report 98-12.
INSTITUTION Twente Univ., Enschede (Netherlands). Faculty of Educational Science and Technology.
PUB DATE 1998-00-00
NOTE 33p.
AVAILABLE FROM Faculty of Educational Science and Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
PUB TYPE Reports - Evaluative (142)
EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Higher Education; *Item Banks; *Selection; *Test Construction; Test Items
IDENTIFIERS *Integer Programming; Law School Admission Test

ABSTRACT

Six methods for assembling tests from a pool with an item-set structure are presented. All methods are computational and based on the technique of mixed integer programming. The methods are evaluated using such criteria as the feasibility of their linear programming problems and their expected solution times. The methods are illustrated for two item pools with a set structure from the Law School Admission Test (LSAT). The methods are: (1) simultaneous selection of items and sets; (2) simultaneous selection with pivot items; (3) all items per set selected; (4) decision variables for subsets (power set approach); (5) two-stage selection; and (6) two-stage selection (alternative version). (Contains 3 tables, 2 figures, and 12 references.) (Author/SLD)

Optimal Assembly of Tests with Item Sets

Research Report 98-12

Wim J. van der Linden

1

faculty of
EDUCATIONAL SCIENCE AND TECHNOLOGY

University of Twente

Department of
Educational Measurement and Data Analysis

2

Optimal Assembly of Tests with Item Sets

Wim J. van der Linden

University of Twente

Send requests for information to: W.J. van der Linden, Department of Educational Measurement and Data Analysis, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands. Email: vanderlinden@edte.utwente.nl

Abstract

Six methods for assembling tests from a pool with an item-set structure are presented. All methods are computational and based on the technique of mixed integer programming. The methods are evaluated using such criteria as the feasibility of their linear programming problems and their expected solution times. The methods are illustrated for two item pools with a set structure from the Law School Admission Test (LSAT).

Optimal Assembly of Tests with Item Sets

A well-known format in achievement testing is the one of a test with sets of items related to a common stimulus. The format has been ubiquitous in testing of reading comprehension where examinees are typically offered a series of text passages each followed by a set of questions on them. Other examples can be found in testing of achievements in science when sets of items relate to a description of a common data set or experiment, or in law exams with sets of questions addressing a common lawsuit. The use of tests with an item-set structure has become popular lately as a result of the trend to making testing more performance based.

Assembling tests from an item pool with a set structure tends to be much more complicated than from a pool of self-contained items, mainly because they have to obey more complicated lists of specifications. For example, specifications for test with item sets do not only involve constraints on item and test attributes but also on stimulus attributes as well as on distributions of item attributes in items sets. In addition, this type of test assembly has to meet the following logical or Boolean constraints:

(1) if any of the items in a set is selected, its stimulus is selected;

(2) if any of the items in a set is selected, a minimum and/or maximum number of the items in the set is selected.

This paper presents a number of methods for assembling tests from pools with items sets. All methods are computational and based on the technique of mixed integer programming (LP). The technique will be briefly introduced in the description of the first method below. A more general introduction to LP-based test assembly and a review of its current applications are given in van der Linden (1998).

It is assumed that test assembly is IRT based, that is, its objective is to assemble a test with an information function that has to meet a given target (Birnbaum, 1968). In the empirical examples in this paper, the 3-parameter logistic (3-PL) model is assumed to hold. The response function for item i in this model is given by:

$$p_i(\theta) \equiv \text{Prob}\{U_i=1|\theta\} \equiv c_i + (1-c_i)\frac{\exp[a_i(\theta-b_i)]}{1+\exp[a_i(\theta-b_i)]}, \quad (1)$$

where $\theta\in(-\infty,\infty)$ is the examinee parameter, $b_i\in(-\infty,\infty)$ is the item difficulty, $a_i\in[0,\infty)$ is the item discrimination, and $c_i\in[0,1]$ a parameter needed to deal with guessing on the item . The question if pools with item sets are likely to fit the model in Equation 1 is deliberately omitted here (for this question, see Rosenbaum, 1987).

The paper is organized as follows. First, the various types of constraints on item selection possible in test assembly with item sets are described. Then six different methods and their associated mixed integer programing models for test assembly subject to such constraints are introduced. The methods are evaluated using such criteria as the feasibility of their LP problem and their expected solution times. The final section of the paper presents some empirical examples in which the results for these methods are compared for two item pools from the Law School Admission Test (LSAT).

Constraints on Tests Assembly with Item Sets

Specifications for tests with item sets typically address attributes defined at three different levels in the test (individual items; sets; complete test). In addition, they imply item-selection constraints on attributes at their primary level but often also at higher levels of aggregation. As an example of the distinction between attribute level and constraint level, consider the following specification:

> "No item set in the test should have more than two items with a multiple-choice format."

This specification addresses an attribute defined at item level ("response format") but involves a constraint on this attribute at the level of the item sets ("no more than two multiple-choice items per set").

The following classifications of attribute and constraint level are used to formulate the test assembly methods later in this paper:

Attribute Level.

Three different attribute levels are distinguished. At each of these levels several types of attributes can be met. However, in practice some types of attributes are more likely to occur at certain levels than others. The attribute levels addressed in this paper are:

1. Item level. Examples of item attributes are: content, cognitive level, values of statistical parameters, format, and word counts. Some of these attributes are categorical, that is, imply a partition of the item pool with each class representing a categorical value of the attribute (e.g., "response format" with values "multiple-choice" and "constructed response"); others are quantitative (e.g., item p-values). Both types of attributes lead to different types of constraints (for examples, see Equations 5-8 below).
2. Stimulus level. Stimuli can have the same kind of categorical attributes as items (content; cognitive level; etc). However, except for an attribute as word counts, they are unlikely to have quantitative attributes associated with them, In particular, they seldom have statistical attributes.
3. Test level. Attributes can also be defined at the level of the complete test. Examples are: test length, maximum distance between the test information function and a target, and (classical) reliability of the test. Attributes at this level are generally quantitative and statistical by nature.

Constraint Levels

Constraints at four different levels are distinguished. Each constraint level addresses attributes defined at the same level or aggregates of attributes defined at a lower level in the test. The levels considered in this paper are:

1. Item level. Constraints at item level generally stipulate the inclusion or exclusion of items with certain attribute values from the test. Example of constraints formulated at item level are:

"Each items should be on analytic reasoning";

"No completion items should be used".

2. Item-set level. Constrains at item-set level control the distribution of the values of categorical item attributes, require a function of the values of quantitative item attributes to be between bounds, or require the simultaneous occurrence of certain item and stimulus attributes. Examples of these types of constraints are:

   "Each item set should have at least two items on applications";

   "The average p-value of the first item set should not be smaller than .60;

   "Item sets with a stimulus describing a physics experiment should have no more than two items with graphical information."

3. Stimulus level. Just as at item level, constraints at stimulus level govern the inclusion or exclusion of stimuli with certain attribute values from the test. An example of a constraint at stimulus level is:

   "No stimulus with should have a word count larger than 350 words".

4. Test level. Constraints at test level apply either to test attributes or to distributions or functions of values of item or stimulus attributes. Examples of constraints at this level are:

   "The test should have three stimuli presenting a recent newspaper article";

   "The test information function should be uniform over the interval from θ=-2.0 and 1.5".

As already noted, the above classification of constraint levels implies a hierarchical

structure with respect to the attribute levels. Each constraint is formulated at the same level as the attributes it addresses or higher, but never lower. In fact, attributes themselves may have a hierarchical structure too, in particular if they are quantitative. Examples of such attributes are test information function and the classical reliability coefficient; both are defined as mathematical functions of lower-level attributes (item information, p-values and covariances between items).

Finally, it is observed that the two classifications may have to be extended with an intermediate level when the test has subtests or sections. Likewise, higher levels may have to be added, for example, when a set of parallel test forms or a set of tests for use in multi-stage testing is assembled.

Methods of Test Assembly

Six different methods for assembling tests with item sets are presented. Some of these methods are exact; others require manual preprocessing of the item pool or have a heuristic element. The features of these methods will be evaluated against each other after the methods have been described.

Method 1: Simultaneous Selection of Items and Sets

The key feature of this method is that separate decision variables for the selection of items and stimuli are defined. The variables are used to model the constraints to be imposed on the selection of items and stimuli. Special constraints are added to keep the selection of items and stimuli consistent, that is, prevent that items (stimuli) are selected but their stimuli (items) are not. This first method was introduced in van der Linden (1992).

Let the stimuli in the pool be indexed by s=1,...,S, and the items nested under stimulus s by i_s=1,...,I_s. Variables z_s are used to select the stimuli; they take the value 1 if stimuli s is selected for the test and the value 0 otherwise. Likewise, 0-1 variables x_{i_s} are defined for the decision on item i_s.

It is assumed that target values, $T(\theta_k)$, k=1,...,K, are specified for the value of the

test information function at θ_k. The value of the information for item i_s at θ_k is denoted as $I_{i_s}(\theta_k)$. In the model below, for each value θ_k the test information function is required to be in the interval $(T(\theta_k)-y,\ T(\theta_k)+y)$, where y is a (real-valued) variable defining the size of the interval. The objective of the decision problem is to minimize y. For a more extensive description of this minimax objective, see van der Linden and Boekkooi-Timminga (1989).

In addition, the following notation is needed:

q_{i_s}: value of item i_s on quantitative attribute q;

r_s: value of stimulus s on quantitative attribute r;

C_g: set of indices of items with value g on categorical attribute C, g=1,...,G;

D_h: set of indices of stimuli with value h on categorical attribute D, h=1,...,H;

n: number of items in the test.

The following model for simultaneous selection of items and stimuli is presented:

minimize y (2)

subject to

$$\sum_{s=1}^{S} \sum_{i_s=1}^{I_s} I_{i_s}(\theta_k) x_{i_s} + y \geq T(\theta_k), \quad k=1,...,K \qquad \text{(test information function)} \quad (3)$$

$$\sum_{s=1}^{S} \sum_{i_s=1}^{I_s} I_{i_s}(\theta_k) x_{i_s} - y \leq T(\theta_k), \quad k=1,...,K \qquad \text{(test information function)} \quad (4)$$

$$\sum_{s=1}^{S} \sum_{i=1}^{I_s} q_{i_s} x_{i_s} \geq q^{(l)} \qquad \text{(quantitative item attribute)} \quad (5)$$

$$\sum_{s=1}^{S} \sum_{i=1}^{I_s} q_{i_s} x_{i_s} \leq q^{(u)} \qquad \text{(quantitative item attribute)} \quad (6)$$

$$\sum_{s=1}^{S} \sum_{i_s \in C_g} x_{i_s} \geq n_g^{(l)}, \quad g=1,,,G \qquad \text{(categorical item attribute)} \quad (7)$$

$$\sum_{s=1}^{S} \sum_{i_s \in C_g} x_{i_s} \leq n_g^{(u)}, \quad g=1,,,G \qquad \text{(categorical item attribute)} \quad (8)$$

$$\sum_{s=1}^{S} r_s z_s \geq r^{(l)} \qquad \text{(quantitative stimulus attribute)} \quad (9)$$

$$\sum_{s=1}^{S} r_s z_s \leq r^{(u)} \qquad \text{(quantitative stimulus attribute)} \quad (10)$$

$$\sum_{s=1}^{S} z_s \geq n_h^{(l)}, \quad h=1,,,H \qquad \text{(categorical stimulus attribute)} \quad (11)$$

$$\sum_{s=1}^{S} z_s \leq n_h^{(u)}, \quad h=1,,,H \qquad \text{(categorical stimulus attribute)} \quad (12)$$

$$\sum_{s=1}^{S} z_s = m \qquad \text{(number of item sets)} \quad (13)$$

$$\sum_{i=1}^{I_s} x_{i_s} - n_s^{(l)} z_s \geq 0, \quad s=1,...,S \qquad \text{(number of items per set)} \quad (14)$$

$$\sum_{i=1}^{I_s} x_{i_s} - n_s^{(u)} z_s \leq 0, \quad s=1,...,S \qquad \text{(number of items per set)} \quad (15)$$

$$\sum_{s=1}^{S} \sum_{i_s=1}^{I_s} x_{i_s} - n^{(l)} \geq 0 \qquad \text{(test length)} \quad (16)$$

$$\sum_{s=1}^{S} \sum_{i_s=1}^{I_s} x_{i_s} - n^{(u)} \leq 0 \qquad \text{(test length)} \quad (17)$$

$$y \geq 0 \qquad \text{(definition of decision variable)} \quad (18)$$

$$x_{i_s} \in \{0,1\}, \; i_s=1,...,I_s, \; s=1,...,S \qquad \text{(definition of decision variables)} \quad (19)$$

$$z_s \in \{0,1\}, \quad s=1,...,S \qquad \text{(definition of decision variables)} \quad (20)$$

The constraints in Equations 3 and 4 tighten the values of the test information function to the common interval $(T(\theta_k)-y,\ T(\theta_k)+y)$. The size of the interval is minimized in Equation 2. Equations 5-8 show how sums of values of quantitative attributes or distributions of items across values of categorical attributes can be constrained to meet lower (subscript "l") and upper bounds (subscript "u"). The same is demonstrated for quantitative and categorical stimuli in Equations 9-12. For convenience, examples are given for constraints at test level only. Other constraint levels in the earlier classification can be realized adapting the sums in the equations.

The number of item sets to be selected is set in the constraint in Equation 13. Equations 14 and 15 have a double purpose. On the one hand, they constrain the numbers of items per stimulus, where $n_s^{(l)}$ and $n_s^{(u)}$ are the lower and upper bounds on the number of items in set s, respectively. On the other hand, as can easily verified by substituting a 0 and 1 for z_s, the constraints coordinate the selection of items and stimuli. These constraints are the logical or Boolean constraints needed for test assembly with item sets alluded to earlier. The total number of items in the test is set through Equations 16 and 17.

Equations 18-20 constrain the decision variables to their proper domains of possible values. Observe that y is a decision variable too. Due to its presence, the problem involved in solving Equations 2-20 is known as a mixed integer programming problem. General LP software (e.g. CPLEX; see ILOG, 1998) or one of the algorithms in the test assembly software package ConTEST (Timminga, van der Linden, & Schweizer, 1996) can be used to solve the model for optimal values for the decision variables. Numerical aspects of solving models as in Equations 2-20 will be discussed further below.

Method 2: Simultaneous Selection with Pivot Items

In mixed integer programming, solution times generally depend on the numbers of variables in the model. It is therefore advantageous to find models for test assembly problems with item sets that are based on fewer variables with results that closely

approximate those for a full simultaneous approach.

A reduction of the number of variables is possible by assigning one item in each item set in the pool the special status of "pivot item". Formally, a pivot item is defined as an item selected for the test if and only if its stimulus is selected for the test. In practice, test specialists can be asked to select as pivot items the ones they feel represent their stimuli best and would be their first option if the test were to be assembled by hand. Of course, what is "best" should follow from the specifications for the test in combination with the relative scarcity of the item attributes in the pool.

Because the decision variables for pivot items and stimuli have identical values in any solution, the decision variables for the pivot items can be used as carriers for the attributes of the stimuli and to formulate constraints on stimulus selection. Hence, if pivot items have been selected, no separate decision variables for the stimuli are needed.

Let i_s^* be the index value of the pivot item for stimulus s. The only thing needed to change the model in Equations 2-20 into a model for Method 2 is:

1. Substitution of decision variables $x_{i_s^*}$ for decision variables z_s.
2. Omission of the constraints in Equation 20.

Observe that the constraints in Equations 14 and 15 now guarantee that pivot items are selected any time a sufficient number of items for their stimulus is. These constraints thus provide the formal definition of the status of the pivot items.

Method 3: All Items Per Set Selected

In the previous method, the number of decision variables was reduced by removing the variables for the stimuli from the model. A more dramatic reduction is possible if the decision variables for the items can be removed. This possibility arises if the numbers of items per stimulus in the pool meet the specifications for the test, for example, when the pool has to serve only one testing program and the item sets in the pool have been tailored to the specifications for this test. Another application arises if all items sets are edited by test specialists prior to the test assembly process removing the worst items from the sets

until their size meets the specifications.

In either application, the only decisions left are which stimuli to select for the test. As all items in the sets are selected along with their stimulus, aggregated values of the item attributes in the sets can be assigned as attributes to the stimuli, and the decision variables for the stimuli can be used to formulate constraints on the item attributes.

In the model in Equations 2-20, item attributes were used in Equations 2 and 3 (information function values), Equations 6 and 7 (quantitative attributes), and Equations 8 and 10 (categorical attributes). Constraints on the selection of the items were also formulated in Equations 14-17. Let

$$n_s \equiv \text{number of items in set s;} \tag{21}$$

$$c_{sg} \equiv \text{number of item s in set s with index in } C_g; \tag{22}$$

$$I_s(\theta_k) \equiv \sum_{i_s=1}^{I_s} I_{i_s}(\theta_k) \qquad \text{(item set information)} \tag{23}$$

$$q_s \equiv \sum_{i_s=1}^{I_s} q_{i_s} \qquad \text{(sum of values on quantitative attribute)} \tag{24}$$

The model for Method 3 is derived from the one in Equations 2-20 by making the following modifications:

1. Equations 3-8 and 16-17 are reformulated as:

$$\sum_{s=1}^{S} I_s(\theta_k) z_s + y \geq T(\theta_k), \quad k=1,...,K$$

(test information function) (25)

$$\sum_{s=1}^{S} I_s(\theta_k) z_s - y \leq T(\theta_k), \quad k=1,...,K$$

(test information function) (26)

$$\sum_{s=1}^{S} q_s z_s \geq q^{(l)}$$ (quantitative item attribute) (27)

$$\sum_{s=1}^{S} q_s z_s \leq q^{(u)}$$ (quantitative item attribute) (28)

$$\sum_{s=1}^{S} c_{sg} z_s \geq n_g^{(l)}, \quad g=1,,,G$$ (categorical item attribute) (29)

$$\sum_{s=1}^{S} c_{gs} z_s \leq n_g^{(u)}, \quad g=1,,,G$$ (categorical item attribute) (30)

$$\sum_{s=1}^{S} n_s z_s - n^{(l)} \geq 0$$ (test length) (31)

$$\sum_{s=1}^{S} n_s z_s - n^{(u)} \leq 0$$ (test length) (32)

2. The constraints on numbers of items per set in Equations 14 and 15 and the definition of the decision variables for the items in Equation 19 are removed from the model.

Method 4: Decision Variables for Subsets (Power Set Approach)

The following method was inspired by an observation in Swanson and Stocking (1993, p. 157). If the number of items in set s is equal to n_s the maximum number of (nonempty) different sets in the test selected from s is equal to $2^{n_s}-1$, that is, the number of elements in the power set of s minus the null set. Assembling the test can be modeled using separate decision variables for each subset and without any variable for the items.

Let z_{p_s}, $p=1,...,2^{n_s}$, be the pth element in the power set of item set s=1,..,S. For each element in the power set, definitions for the numbers of items and quantitative and

categorical item attributes as in Equations 21-24 are introduced. The model needed to implement a power set approach set is analogous to the one for the previous case. The only exceptions are:

1. The addition of the following set of constraints to prevent selection of more than one subset per item set:

$$\sum_{p=1}^{2^{n_s}} z_{p_s} \leq 1, \quad s=1,...,S \quad \text{(mutually exclusive subset selection)} \quad (33)$$

2. The replacement of the constraint on the number of item sets to be selected in Equation 13 by:

$$\sum_{s=1}^{S} \sum_{p=1}^{n_s} z_{p_s} = m \quad \text{(number of item sets)} \quad (34)$$

Observe that the constraint in Equation 34 works correctly only in combination with the ones in Equation 33.

This method yields an optimal solution. However, its number of variables easily becomes large. In fact, the method is practical only when some of the item sets in the pool have one or two items too many. In all other cases, Method 1 is superior in the sense that it also produces an optimal result but has fewer variables.

Method 5: Two-Stage Selection

If a mathematical programming problem is too large, an obvious approach is to approximate the problem by a series smaller problems. This strategy is followed in Method 5 which is based on two stages: In Stage 1 item sets are selected, whereas in Stage 2 the test is assembled from the sets selected in Stage 1.

The model for Stage 1 is identical to the one for Method 3, with the following modifications of the constraints in Equations 26-32:

1. Use of the constraints with upper bounds on categorical item attributes and

test length in Equations 30 and 32 can postponed to Stage 2. However, the versions of these constraints with the lower bounds are kept to maximize the likelihood of a feasible result in Stage 2.

2. Constraints on quantitative item attributes are rescaled at item level. For example, the constraints on test information in Equations 25 and 26 can be reformulated as:

$$\sum_{s=1}^{S} n_s^{-1} I_s(\theta_k) z_s + y \geq n^{-1} T(\theta_k), \quad k=1,...,K$$

(test information function) (35)

$$\sum_{s=1}^{S} n_s^{-1} I_s(\theta_k) z_s - y \leq n^{-1} T(\theta_k), \quad k=1,...,K$$

(test information function) (36)

where n is the intended test length. This rescaling is necessary to maximize the likelihood of a good fit of the information function for the items selected in Stage 2. If test length is constrained by different upper and lower bounds, the mean of these two values can be chosen as the value of n in Equations 35 and 36.

The model for Stage 2 is identical to the one for Method 1 in Equations 2-20, with the following modifications:

1. The constraints on quantitative and categorical stimulus attributes in Equations 9-12 have already been realized at Stage 1 and are no longer needed.
2. The constraints in Equations 14-15 are replaced by

$$\sum_{i=1}^{I_s} x_{i_s} \geq n_s^{(l)}, \quad s=1,\ldots,S^* \qquad \text{(number of items per set)} \quad (37)$$

$$\sum_{i=1}^{I_s} x_{i_s} \leq n_s^{(u)}, \quad s=1,\ldots,S^* \qquad \text{(number of items per set)} \quad (38)$$

where s now runs over the item sets selected in Stage 1.

3. The definition of the decision variables in Equation 20 is no longer needed.

Method 6: Two-Stage Selection (Alternative Version)

The previous method has the advantage of a small number of decision variables but runs the danger of a result in Stage 1 that overconstrains the selection space in Stage 2. A potential useful alternative to the previous method is therefore to select a larger number of item sets in Stage 1 than actually needed in Stage 2. In fact, Stage 1 can be used just to weed out item sets form the pool unlikely to be selected in Stage 2.

The model needed for Stage 1 is identical to the one for this stage in Method 5. The only difference is the number of item sets selected in Equation 13. The model for Stage 2 is identical to the one of simultaneous selection of items and stimuli in Method 1. The model is now defined only over the part of the pool selected in Stage 1.

Discussion

Method 1 is based on the most general formulation of the test assembly problem. Its implementation does not require any manual preprocessing of the item pool. Also, it produces an optimal solution, provided the solution can be found in realistic time.

Method 2 and 3 are reductions of the original problem based on previous assignment of pivot items in the sets and reduction of the size of the item sets by weeding out their worst items. However, the reduction in the number of variables should be evaluated against the fact that the quality of the solution depends on the results of the preprocessing of the item pool. If wrong selections are made at this stage, the solution, though optimal in the reduced problem, may be suboptimal in the original problem.

Method 4 is a generalization of Method 3 in the sense that it has decision variables

associated not only with the item sets but also with each of their subsets. Like Method 1, Method 4 produces an optimal result to the original problem. However, since the number of variables in Method 4 increases dramatically as a function of the difference between the size of the item sets in the pool and the size requested for the test, it may have much more difficulty finding a solution in realistic time than for Method 1.

The advantage of Method 5 and 6 is that they involve two small problems that can be solved quickly for item pools of a realistic size. Also, unlike Method 2 and 3, these methods do not involve any manual preprocessing of the item pool. A potential disadvantage of these methods is the possibility of a solution in Stage 1 that does not allow a feasible test at Stage 2. Method 6 is expected to perform generally better in this respect due to its less stringent selection in Stage 1.

Empirical Examples

The methods were applied to the problem of assembling the two sections of the LAST that have an item-set structure. The sections are coded here as SA and SB. (The LSAT has a third section that does not have item sets.) The numbers of items and stimuli in these two sections and their item pools are given in Table 1.

[Table 1 about here]

For both sections of the LSAT, models were formulated for Methods 1-3 and 5-6. For Method 2 and 3, LSAT specialists selected the pivot items and reduced the item sets in the pools to appropriate lengths. Method 6 was implemented by selecting twice as many items sets in Stage 1 as needed in Stage 2. The models dealt with such attributes as item and stimulus types (several levels), possible gender and minority orientation of item sets, answer key distributions of the items, and word counts of the stimuli. The numbers of variables and constraints in the models for these two sections are given in Table 2. It

[Table 2 about here]

reminded that the number of variables in Method 5 and 6 for Stage 2 depend on the items sets selected in Stage 1.

Method 4 was omitted because of its large number of decision variables. For example, for the SA pool a typical item set has 11 items whereas only 5-7 items per set are needed in the test. For this set only the number of variables would have been equal to $\binom{11}{5}+\binom{11}{6}+\binom{11}{7}=1254$.

The target information functions for SA and SB are shown in Figures 1 and 2. For

[Figures 1-2 about here]

all methods the models constrained the test information functions at θ=-1.8, -0.9, 0.0, 0.9, and 1.8. Solutions to the models were obtained using the branch-and-bound algorithm as implemented in CPLEX (ILOG, 1998) on a PC with Pentium Pro 166MHz processor. The algorithm was stopped as soon as the differences between the test information and target values were smaller than 3% of the lowest target value. Since the lowest target value for SA was .8892 at θ=-1.8, the stopping criterion in this case was a maximum difference smaller than .08x.8892=.03. For SB, the smallest target value and stopping criterion were 2.0796 and .06, respectively. Because the objective function in Equation 2 is the largest difference between the test information function and target values over all θ values, the stopping criterion could be applied directly to value of this function.

Table 3 gives some technical results for these two series of examples. All methods

[Table 3 about here]

immediately produced feasible solutions for the two sections. The only exception was the combination of Method 6 and SB. In Stage 1, this method selected a combination of item sets that did not contain a feasible combination of sets for Stage 2. However, relaxing one of the constraints on the item sets, replacing "=2" by "≤3", did produce a solution. The CPU times for all method were satisfactory. Methods 1 and 2 had the largest numbers of variables and were slowest. Surprisingly, the small reduction of the numbers of variables in Method 2 realized by introducing pivot items did not pay off in a smaller CPU time but the reduction in Method 4 had a dramatic effect. In fact, all methods based on a larger reduction of variables or a two-stage implementation of the selection procedure were very

quick (generally less than 1 second of CPU time). The last column in Table 3 shows the values of the objective function for the solution, y^*. As already noticed, these values are equal to the largest difference between the test information function values and their target values across the θ values used in the models. Methods 1 and 2 produced the best results, immediately followed by Method 6 for SB. The other methods produced larger differences.

A graphical presentation of the results is offered in Figures 1-2. For SA and Method 1, 2 and 3 the test information functions were close to the target function. For SB the best results were obtained for Methods 1, 2 and 6; the test information functions for these methods were virtually indistinguishable from the target function. Also, Methods 3 and 5, though not satisfactory, performed considerably better for SB than the two worst performing methods for SA. Observe that, both for SA and SB, Method 6, which constrains the item set selection in Stage 1 less stringently, did better indeed than Method 5.

Concluding Remark

The empirical results in this paper are offered only as an example. Though most results were as expected, a surprise was the fact that Method 2 and 3 outperformed Method 5 and 6 for SA whereas the opposite tendency was observed for SB. These results show the dependency of the performance of test assembly methods on the composition of the item pool. When generalizing the results in these examples to other applications, this dependency should be taken into account.

References

Adema, J.J. (1992). Methods and models for the construction of weakly parallel tests. Applied Psychological Measurement, 16, 53-63.

Armstrong, R.D., Jones, D.H., & Wang, Z. (1995). Network optimization in constrained standardized test construction. In K.D. Lawrence Ed.). Applications of management science: Network optimization applications (Vol. 8, pp. 189-212). Greenwich, CT: JAI Press.

Birnbaum, A. (1986). Some latent trait models and their use in inferring an examinee's ability. In F.M. Lord and M.R. Novick, Statistical theories of mental test scores. Reading, MA: Addison-Wesley.

Boekkooi-Timminga, E. (1987a). Simultaneous test construction by zero-one programming. Methodika, 1, 1101-112.

ILOG (1998). CPLEX 6.0 Documentation supplement [Computer software]. Incline Village, NV: ILOG, Inc.

Rosenbaum, P.R. (1987). A note on item bundles. Psychometrika, 53, 349-360.

Swanson, L. & Stocking, M.L. (1993). A model and heuristic for solving very large item selection problems. Applied Psychological Measurement, 17, 151-166.

Timminga, E. & van der Linden, W.J., & Schweizer, D.A. (1996). ConTEST 2.0: A decision support system for item banking and optimal test assembly [Computer software and manual]. Groningen, The Netherlands: iec ProGAMMA.

van der Linden, W.J. (1992). Selecting passage-based items for achievement tests [Internal report]. Iowa City, IA: American College Testing.

van der Linden, W.J. (1998). Optimal assembly of psychological and educational tests. Applied Psychological Measurement, 22, 259-270. [With a bibliography]

van der Linden, W.J., & Adema, J.J. (1998). Simultaneous assembly of multiple test forms. Journal of Educational Measurement, 35, in press.

van der Linden, W.J., & Boekkooi-Timminga, E. (1989). A maximin model for test design with practical constraints. Psychometrika, 54, 237-247.

Authors' Note

This study received funding from the Law School Admission Council (LSAC). The opinions and conclusions contained in this paper are those of the authors and do not necessarily reflect the position or policy of LSAC. The author is indebted to Stephen W. Luebke for providing the data for the emprical examples and Wim M.M. Tielen for his computational support.

Table 1

Numbers of Items and Stimuli in Pools for SA and SB

	Pool			Test		
Section	#Items	#Stimuli	#Items/Stimulus	#Items	#Stimuli	#Items/Stimulus
SA	208	24	5-11	22-24	4	5-7
SB	240	24	8-12	26-28	4	5-8

Table 2

Numbers of Variables and Constraints in Models for Five Test Assembly Methods

Method	Section	#Variables	#Constraints
1	SA	233	91
	SB	265	109
2	SA	209	91
	SB	241	109
3	SA	25	41
	SB	25	60
5	SA-1[1)]	25	29
	SA-2	33[2)]	36
	SB-1	25	37
	SB-2	37[2)]	58
6	SA-1	25	29
	SA-2	74[2)]	57
	SB-1	25	37
	SB-2	88[2)]	75

Notes: 1. Second code indicates stage; 2. Number is dependent on output from Stage 1.

Table 3

Technical Results for Five Test Assembly Methods

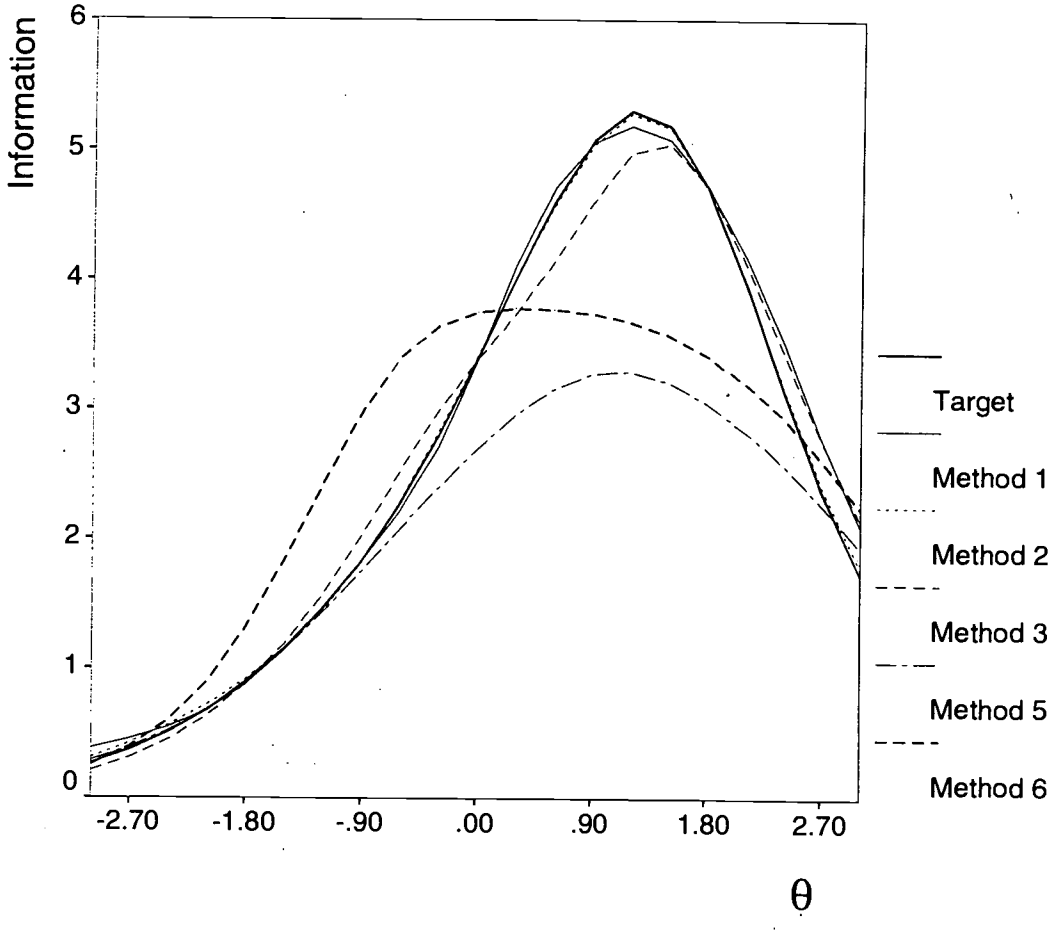
Method	Section	Feasibility	CPU Time	y^*
1	SA	+	4-5 mins	.021
	SB	+	1-2 mins	.011
2	SA	+	20 mins	.032
	SB	+	90 mins	.064
3	SA	+	<1 sec	.473
	SB	+	<1 sec	1.099
5	SA-1[1)]	+	<1 sec	.232
	SA-2	+	<1 sec	1.801
	SB-1	+	<1 sec	.432
	SB-2	+	<1 sec	.881
6	SA-1	+	<1 sec	.838
	SA-2	+	<1 sec	1.339
	SB-1	+	<1 sec	1.152
	SB-2	-[2)]	1-2 secs	.049

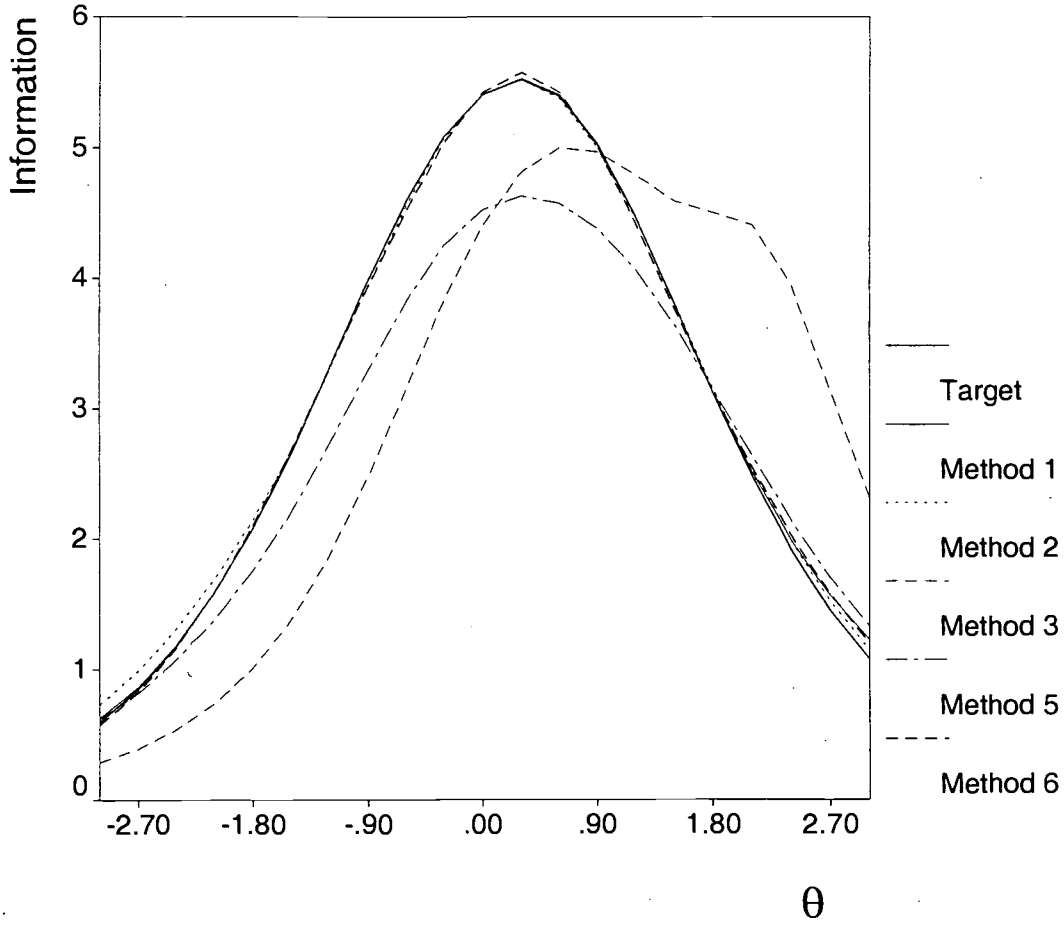
Notes: 1. Second code indicates stage; 2. CPU time and value of y were obtained after relaxation of one constraint to get a feasible solution.

Figure Captions

Figure 1. Target information function and test information functions for Method 1, 2, 3, 5 and 6 (Section SA).

Figure 2. Target information function and test information functions for Method 1, 2, 3, 5 and 6 (Section SB).


Information
6
5
4
3
2
1
0
-2.70
-1.80
-.90
.00
.90
1.80
2.70
θ
Target
Method 1
Method 2
Method 3
Method 5
Method 6


Information
6
5
4
3
2
1
0
-2.70
-1.80
-.90
.00
.90
1.80
2.70
θ
Target
Method 1
Method 2
Method 3
Method 5
Method 6

Titles of Recent Research Reports from the Department of Educational Measurement and Data Analysis. University of Twente, Enschede, The Netherlands.

RR-98-12 W.J. van der Linden, *Optimal Assembly of Tests with Item Sets*

RR-98-11 W.J. van der Linden, B.P. Veldkamp & L.M. Reese, *An Integer Programming Approach to Item Pool Design*

RR-98-10 W.J. van der Linden, *A Discussion of Some Methodological Issues in International Assessments*

RR-98-09 B.P. Veldkamp, *Multiple Objective Test Assembly Problems*

RR-98-08 B.P. Veldkamp, *Multidimensional Test Assembly Based on Lagrangian Relaxation Techniques*

RR-98-07 W.J. van der Linden & C.A.W. Glas, *Capitalization on Item Calibration Error in Adaptive Testing*

RR-98-06 W.J. van der Linden, D.J. Scrams & D.L.Schnipke, *Using Response-Time Constraints in Item Selection to Control for Differential Speededness in Computerized Adaptive Testing*

RR-98-05 W.J. van der Linden, *Optimal Assembly of Educational and Psychological Tests, with a Bibliography*

RR-98-04 C.A.W. Glas, *Modification Indices for the 2-PL and the Nominal Response Model*

RR-98-03 C.A.W. Glas, *Quality Control of On-line Calibration in Computerized Assessment*

RR-98-02 R.R. Meijer & E.M.L.A. van Krimpen-Stoop, *Simulating the Null Distribution of Person-Fit Statistics for Conventional and Adaptive Tests*

RR-98-01 C.A.W. Glas, R.R. Meijer, E.M.L.A. van Krimpen-Stoop, *Statistical Tests for Person Misfit in Computerized Adaptive Testing*

RR-97-07 H.J. Vos, *A Minimax Sequential Procedure in the Context of Computerized Adaptive Mastery Testing*

RR-97-06 H.J. Vos, *Applications of Bayesian Decision Theory to Sequential Mastery Testing*

RR-97-05 W.J. van der Linden & Richard M. Luecht, *Observed-Score Equating as a Test Assembly Problem*

RR-97-04 W.J. van der Linden & J.J. Adema, *Simultaneous Assembly of Multiple Test Forms*

RR-97-03 W.J. van der Linden, *Multidimensional Adaptive Testing with a Minimum Error-Variance Criterion*

RR-97-02 W.J. van der Linden, *A Procedure for Empirical Initialization of Adaptive Testing Algorithms*

RR-97-01 W.J. van der Linden & Lynda M. Reese, *A Model for Optimal Constrained Adaptive Testing*

RR-96-04 C.A.W. Glas & A.A. Béguin, *Appropriateness of IRT Observed Score Equating*

RR-96-03 C.A.W. Glas, *Testing the Generalized Partial Credit Model*

RR-96-02 C.A.W. Glas, *Detection of Differential Item Functioning using Lagrange Multiplier Tests*

RR-96-01 W.J. van der Linden, *Bayesian Item Selection Criteria for Adaptive Testing*

RR-95-03 W.J. van der Linden, *Assembling Tests for the Measurement of Multiple Abilities*

RR-95-02 W.J. van der Linden, *Stochastic Order in Dichotomous Item Response Models for Fixed Tests, Adaptive Tests, or Multiple Abilities*

RR-95-01 W.J. van der Linden, *Some decision theory for course placement*

RR-94-17 H.J. Vos, *A compensatory model for simultaneously setting cutting scores for selection-placement-mastery decisions*

RR-94-16 H.J. Vos, *Applications of Bayesian decision theory to intelligent tutoring systems*

RR-94-15 H.J. Vos, *An intelligent tutoring system for classifying students into Instructional treatments with mastery scores*

RR-94-13 W.J.J. Veerkamp & M.P.F. Berger, *A simple and fast item selection procedure for adaptive testing*

RR-94-12 R.R. Meijer, *Nonparametric and group-based person-fit statistics: A validity study and an empirical example*

RR-94-10 W.J. van der Linden & M.A. Zwarts, *Robustness of judgments in evaluation research*

RR-94-9 L.M.W. Akkermans, *Monte Carlo estimation of the conditional Rasch model*

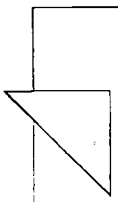
RR-94-8 R.R. Meijer & K. Sijtsma, *Detection of aberrant item score patterns: A review of recent developments*

RR-94-7 W.J. van der Linden & R.M. Luecht, *An optimization model for test assembly to match observed-score distributions*

RR-94-6 W.J.J. Veerkamp & M.P.F. Berger, *Some new item selection criteria for adaptive testing*

...

Research Reports can be obtained at costs, Faculty of Educational Science and Technology, University of Twente, Mr. J.M.J. Nelissen, P.O. Box 217, 7500 AE Enschede, The Netherlands.



faculty of
EDUCATIONAL SCIENCE AND TECHNOLOGY

A publication by
The Faculty of Educational Science and Technology of the University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

TM029499

NOTICE

REPRODUCTION BASIS

☑ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").